

10/043.496

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	0	((('register' same (backup or rocovery)) same (reuse or use)) and (synchroniz\$5 near4 state) and 'compiler' and (@ad<"20020110" or @rlad<"20020110" or @prad<"20020110"))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/06/08 09:32
L2	4	((('register' same (backup or rocovery)) same (write or use)) and (synchroniz\$5 near4 state) and 'compiler' and (@ad<"20020110" or @rlad<"20020110" or @prad<"20020110"))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/06/08 10:03
L3	4	((('register' same (backup or rocovery)) same (write or rename)) and (synchroniz\$5 near4 state) and 'compiler' and (@ad<"20020110" or @rlad<"20020110" or @prad<"20020110"))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/06/08 10:04
L4	0	((('register' same (backup or rocovery)) same (write or rename)) and (synchroniz) and 'compiler' and (@ad<"20020110" or @rlad<"20020110" or @prad<"20020110"))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/06/08 10:04
L5	13	((('register' same (backup or rocovery)) same (write or rename)) and (synchronization) and 'compiler' and (@ad<"20020110" or @rlad<"20020110" or @prad<"20020110"))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/06/08 10:07
L6	11	((('register' same (backup or rocovery)) same (write or rename)) and (map\$4 near3 registe\$2) and 'compiler' and (@ad<"20020110" or @rlad<"20020110" or @prad<"20020110"))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/06/08 10:40
L7	11	('register'.ab. or 'register'.ti.) and ('register' same (backup or recovery)) and 714/2.ccls. and (@ad<"20020110" or @rlad<"20020110" or @prad<"20020110"))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/06/08 10:16

L8	0	((('register' same (backup or rocovery)) same (write or rename)) and ((last near2 use\$2) near3 registe\$2) and 'compiler' and (@ad<"20020110" or @rlad<"20020110" or @prad<"20020110"))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/06/08 10:40
L9	16	('register' same (write or rename)) and ((last near2 use\$2) near3 registe\$2) and 'compiler' and (@ad<"20020110" or @rlad<"20020110" or @prad<"20020110"))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/06/08 10:41


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide



THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Terms used

superscalar processor binary translation processor

Found 40,914 of 155,867

Sort results by


[Save results to a Binder](#)
[Try an Advanced Search](#)

Display results


[Search Tips](#)
[Try this search in The ACM Guide](#)
☐ Open results in a new window

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐

1 [Dynamic translation: Dynamic binary translation for accumulator-oriented architectures](#)

Ho-Seop Kim, James E. Smith

 March 2003 **Proceedings of the international symposium on Code generation and optimization: feedback-directed and runtime optimization**

Full text available:

pdf(1.13 MB)


 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)
[Publisher Site](#)

A dynamic binary translation system for a co-designed virtual machine is described and evaluated. The underlying hardware directly executes an accumulator-oriented instruction set that exposes instruction dependence chains (strands) to a distributed microarchitecture containing a simple instruction pipeline. To support conventional program binaries, a source instruction set (Alpha in our study) is dynamically translated to the target accumulator instruction set. The binary translator identifies ...

2 [Binary translation and architecture convergence issues for IBM system/390](#)

Michael Gschwind, Kemal Ebcioglu, Erik Altman, Sumedh Sathaye

 May 2000 **Proceedings of the 14th international conference on Supercomputing**

Full text available:

pdf(1.44 MB)

 Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

We describe the design issues in an implementation of the ESA/390 architecture based on binary translation to a very long instruction word (VLIW) processor. During binary translation, complex ESA/390 instructions are decomposed into instruction "primitives" which are then scheduled onto a wide-issue machine. The aim is to achieve high instruction level parallelism due to the increased scheduling and optimization opportunities which can be exploited by binary translation software ...

3 [Using Dynamic Binary Translation to Fuse Dependent Instructions](#)

Shiliang Hu, James E. Smith

 March 2004 **Proceedings of the international symposium on Code generation and optimization: feedback-directed and runtime optimization**

Full text available:

pdf(240.50 KB)

 Additional Information: [full citation](#), [abstract](#), [index terms](#)

Instruction scheduling hardware can be simplified and easily pipelined if pairs of dependent instructions are fused so they share a single instruction scheduling slot. We study an implementation of the x86 ISA that dynamically translates x86 code to an underlying ISA that supports instruction fusing. A microarchitecture that is co-designed with the fused instruction set completes the implementation. In this paper, we focus on the dynamic


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

"Register Traffic Analysis for Streamlining Inter-Operation"

THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

 Terms used **Register Traffic Analysis for Streamlining Inter Operation**

Found 7 of 155,867

Sort results by

relevance


[Save results to a Binder](#)
[Try an Advanced Search](#)

Display results

expanded form


[Search Tips](#)
[Try this search in The ACM Guide](#)
☐ Open results in a new window

Results 1 - 7 of 7

 Relevance scale ☐ ☐ ☐ ☐ ☐

1 [Temperature and power aware architectures: Reducing reorder buffer complexity through selective operand caching](#)

Gurhan Kucuk, Dmitry Ponomarev, Oguz Ergin, Kanad Ghose

 August 2003 **Proceedings of the 2003 international symposium on Low power electronics and design**

Full text available: pdf(80.27 KB)

 Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Modern superscalar processors implement precise interrupts by using the Reorder Buffer (ROB). In some microarchitectures, such as the Intel P6, the ROB also serves as a repository for the uncommitted results. In these designs, the ROB is a complex multi-ported structure that dissipates a significant percentage of the overall chip power. Recently, a mechanism was introduced for reducing the ROB complexity and its power dissipation through the complete elimination of read ports for reading out so ...

Keywords: low-complexity datapath, low-power design, reorder buffer, short-lived values

2 [An instruction set and microarchitecture for instruction level distributed processing](#)

Ho-Seop Kim, James E. Smith

 May 2002 **ACM SIGARCH Computer Architecture News**, Volume 30 Issue 2

Full text available: pdf(1.08 MB)

[Publisher Site](#)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

An instruction set architecture (ISA) suitable for future microprocessor design constraints is proposed. The ISA has hierarchical register files with a small number of accumulators at the top. The instruction stream is divided into chains of dependent instructions (strands) where intra-strand dependences are passed through the accumulator. The general-purpose register file is used for communication between strands and for holding global values that have many consumers. A microarchitecture to support ...

3 [Accelerating multi-media processing by implementing memoing in multiplication and division units](#)

Daniel Citron, Dror Feitelson, Larry Rudolph

 October 1998 **Proceedings of the eighth international conference on Architectural support for programming languages and operating systems**, Volume 33, 32 Issue 11, 5